DATA SCIENCE
INTERNSHALA TRAINING

Submitted in partial fulfilment of the
Requirements for the award of
Degree of Bachelor of Technology in Electronics & Communication Engineering

Submitted by:

Name: Shobhit Sharma **Roll No. : 18BT010449**

Name: Ritika Thakur **Roll No. : 18BT010436**

Training period: 30 May 2021 to 13 July 2021

Submitted to: Er. Ankit Sharma

Department of Electronics & Communication Engineering
JAWAHARLAL NEHRU GOVT. ENGINEERING COLLEGE, SUNDERNAGAR, DIST. MANDI (H.P.)

TABLE OF CONTENTS

CERTIFICATE BY COMPANY

INTERNSHALA TRAININGS

Certificate of Training

Ritika Thakur,

student of Jawahar Lal Nehru Govt. Engineering college Sundernagar, has successfully completed a six weeks online training on **Data Science**. The training consisted of Introduction to Data Science, Python for Data Science, Understanding the Statistics for Data Science and Predictive Modeling and Basics of Machine Learning modules.
We wish Ritika all the best for the future.

Sarvesh Agarwal
FOUNDER & CEO, INTERNSHALA

Date of certification: 2021-06-29 Certificate no. : 8819058D-0FBE-5877-50EF-24BC44298760
For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

CERTIFICATE BY COMPANY

INTERNSHALA TRAININGS

Certificate of Training

SHOBHIT SHARMA,

student of Jawaharlal Nehru Government Engineering College, has successfully completed a six weeks online training on **Data Science**. The training consisted of Introduction to Data Science, Python for Data Science, Understanding the Statistics for Data Science and Predictive Modeling and Basics of Machine Learning modules.
In the final assessment, SHOBHIT scored 73% marks.
We wish SHOBHIT all the best for the future.

Sarvesh Agarwal
FOUNDER & CEO, INTERNSHALA

Certificate no. : 96F44981-E5F7-C12E-FE34-FF6F52874B33
For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

DECLARATION BY STUDENT

I hereby declare that the Industrial Training Report entitled "Data Science" is an authentic record of work carried out by me during my training at ―Internshala Trainings from 30 May 2021 to 13 July 2021 for the award of degree of B.Tech. in Electronics & Communication Engineering.

Date: _____ _____

SHOBHIT SHARMA

18BT010449

RITIKA THAKUR

18BT010436

_____

Prof. HIMANSHU MONGA

Head of the Department, ECE

Examined by:

_____

ACKNOWLEDGEMENT

This is my privilege to express my deep sense of gratitude and ineptness to INTERNSHALA and their teachers with whose valuable guidance and encouragement, this training of Data Science has been completed.

I wish my sincere thanks to Head of the Department, ECE for giving us opportunity to do online training.

I am also grateful to all our esteemed faculty of ECE Department for their valuable suggestions and timely help in this training period.

My heartfelt thanks to my friends and classmates for their guidance and timely help.

ABOUT INTERNSHALA

Internshala is an internship and online training platform, based in Gurgaon, India. Founded by Sarvesh Agrawal, an IIT Madras alumnus, in 2010, the website helps students find internships with organisations in India.

HISTORY:

The platform, which was founded in 2010, started out as a WordPress blog that aggregated internships across India and articles on education, technology and skill gap. Internshala launched its online trainings in 2014. As of 2018, the platform had 3.5 million students and 80,000 companies.

PARTNERSHIP:

In August 2016, Telangana's not-for-profit organisation, Telangana Academy for Skill and Knowledge (TASK) partnered with Internshala to help students with internship resources and career services.

In September 2016, Team Indus, Google XPRIZE shortlisted entity has partnered with Internshala for college outreach for its initiative, Lab2Moon.

AWARDS AND RECOGNITION:

In 2011, the website became a part of NASSCOM 10K Startups. In 2015, Internshala was a finalist in People Matters TechHR 2015 Spotlight Awards under 'Futurism in Recruitment' category.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

1. INTRODUCTON

Data Science is a branch of study which involves obtaining meaningful insights from raw & unstructured data. The colossal amount of data is processed through programming, analytics, statistics & predictive modelling. Data Science is a multi- disciplinary field that uses scientific methods, processes, algorithms to produce knowledge & insights from structured & unstructured data. It utilises techniques & theories derived from many fields such as computer science, mathematics, statistics & information science. There will be 60 billion devices connected to internet between 2013-2021 these devices are mobile, laptop, servers, smart watches various CCTVs etc. these devices continuously collecting data and the amount of data generated in this universe increasing exponentially and also we can store this data very cheaply and we can run computations on it with low cost. These trends can demand data scientists for studying this data. The continually increasing access to data is possible due to advancements in technology and collection techniques. Individuals buying patterns and behaviour can be monitored and predictions made based on the information gathered. However, the ever-increasing data is unstructured and requires parsing for effective decision making. This process is complex and time-consuming for companies— hence, the emergence of data science. Data Science is an interdisciplinary field that allows you to extract knowledge from structured or unstructured data. Data science enables us to translate a business problem into a research project and then translate it back into a practical solution.

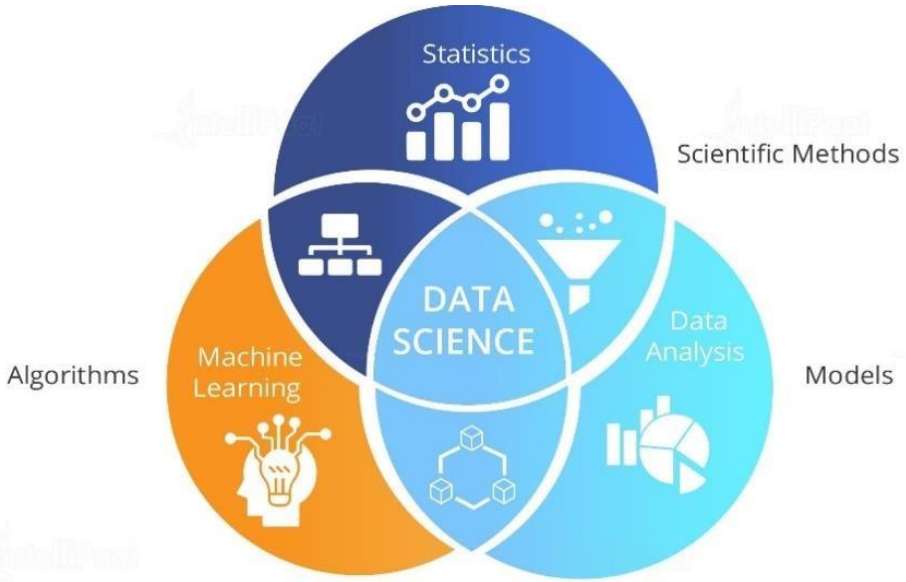


Figure 1 Data Science chart [6]

1.1 HISTORY OF DATA SCIENCE

In 1962, John Tukey described a field he called ―data analysis, which resembles modern data science. Later, attendees at a 1992 statistics symposium at the University of Montpellier II acknowledged the emergence of a new discipline focused on data of various origins and forms, combining established concepts and principles of statistics and data analysis with computing

The term ―data science‖ has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science. In 1996, the International Federation of Classification Societies became the first conference to specifically feature data science as a topic. However, the definition was still in flux. In 1997, C.F. Jeff Wu suggested that statistics should be renamed data science. He reasoned that a new name would help statistics shed inaccurate stereotypes, such as being synonymous with accounting, or limited to describing data. In 1998, Chikio Hayashi argued for data science as a new, interdisciplinary concept, with three aspects: data design, collection, and analysis.

During the 1990s, popular terms for the process of finding patterns in datasets (which were increasingly large) included ―knowledge discovery‖ and ―data mining.

1.2 DATA SCIENCE GOALS AND DELIVERABLES

Here is a short list of common data science deliverables:

- Prediction (predict a value based on inputs)
- Classification (e.g., spam or not spam)
- Recommendations (e.g., Amazon and Netflix recommendations)
- Pattern detection and grouping (e.g., classification without known classes)
- Anomaly detection (e.g., fraud detection)
- Recognition (image, text, audio, video, facial, …)
- Actionable insights (via dashboards, reports, visualizations, …)
- Automated processes and decision-making (e.g., credit card approval)
- Scoring and ranking (e.g., FICO score)
- Segmentation (e.g., demographic-based marketing)
- Optimization (e.g., risk management)
- Forecasts (e.g., sales and revenue)

1.3 DATA SCIENCE COMPONENTS

In data science there are various components we can use for computation of data such as: visualization, statistics, machine learning, deep learning

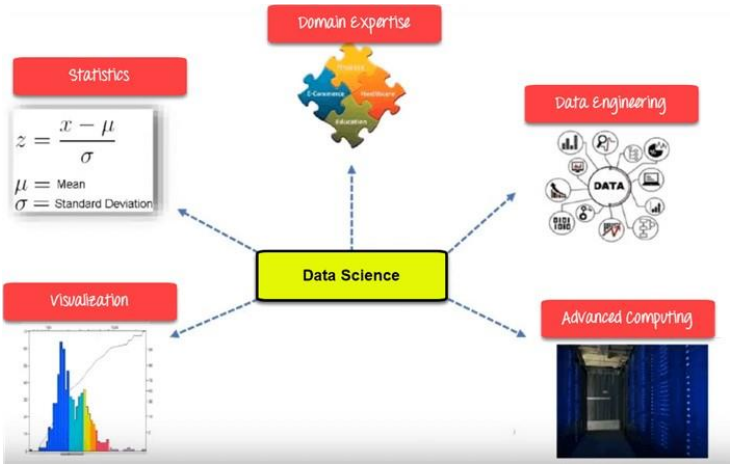


Figure 2 Data Science [7]

- **Statistics**: Statistics is the most critical unit in Data science. It is the method or science of collecting and analysing numerical data in large quantities to get useful insights.

- **Visualization**: Visualization technique helps you to access huge amounts of data in easy to understand and digestible visuals.

- **Machine Learning**: Machine Learning explores the building and study of algorithms which learn to make predictions about unforeseen/future data.

- **Deep Learning**: Deep Learning method is new machine learning research where the algorithm selects the analysis model to follow.

1.4 DATA SCIENCE PROCESS

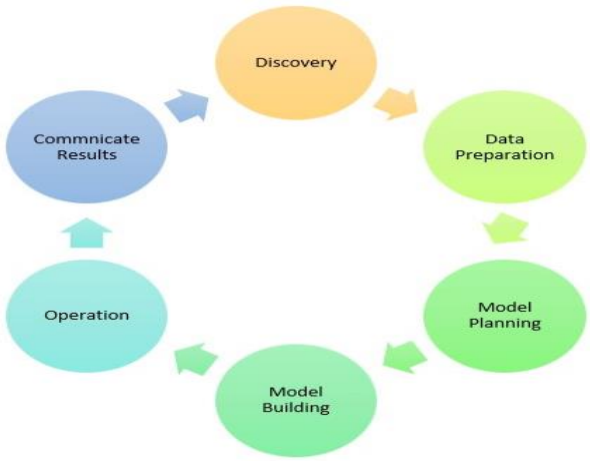


Figure 3 Data Science Process [8]

1. Discovery:

Discovery step involves acquiring data from all the identified internal & external sources which helps you to answer the business question.

The data can be:

- Logs from webservers
- Data gathered from social media
- Census datasets
- Data streamed from online sources using APIs

2. Data Preparation:

Data can have lots of inconsistencies like missing value, blank columns, incorrect data format which needs to be cleaned. You need to process, explore, and condition data before modelling. The cleaner your data, the better are your predictions.

3. Model Planning:

In this stage, you need to determine the method and technique to draw the relation between input variables. Planning for a model is performed by using different statistical formulas and visualization tools. SQL analysis services, R, and SAS/access are some of the tools used for this purpose.

4. Model Building:

In this step, the actual model building process starts. Here, Data scientist distributes datasets for training and testing. Techniques like association, classification, and clustering are applied to the training data set. The model once prepared is tested against the "testing" dataset.

5. Operationalize:

In this stage, you deliver the final baselined model with reports, code, and technical documents. Model is deployed into a real-time production environment after thorough testing.

6. Communicate Results :

In this stage, the key findings are communicated to all stakeholders. This helps you to decide if the results of the project are a success or a failure based on the inputs from the model.

1.5 APPLICATION OF DATA SCIENCE

1. Internet Search:

Google search use Data science technology to search a specific result within a fraction of a second

2. Recommendation Systems:

To create a recommendation system. Example, "suggested friends" on Facebook or suggested videos" on YouTube, everything is done with the help of Data Science.

3. Image & Speech Recognition:

Speech recognizes system like Siri, Google assistant, Alexa runs on the technique of Data science. Moreover, Facebook recognizes your friend when you upload a photo with them, with the help of Data Science.

4. Gaming world:

EA Sports, Sony, Nintendo, are using Data science technology. This enhances your gaming experience. Games are now developed using Machine Learning technique. It can update itself when you move to higher levels.

5. Online Price Comparison:

PriceRunner, Junglee, Shopzilla work on the Data science mechanism. Here, data is fetched from the relevant websites using APIs.

CHAPTER -2

WEEKLY PROGRESS REPORT

WEEK	WEEK START DATE	WEEK END DATE	WEEK OBJECTIVE / TASK
1	30/05/2021	06/06/2021	**INTRODUCTION TO DATA SCIENCE** • Overview of Data Science • Terminologies in Data Science • History & Need of Data Science • Application of Data Science • Scope of Data Science • Instructor Introduction
2	07/06/2021	13/06/2021	**BASIC PYTHON FOR DATA SCIENCE** • Introduction to Python • Understanding Operators • Implementation of operators • Variables and Data Types • Implementation of variable and Data types • Conditional Statements • Implementation of conditional statement • Looping Constructs • Functions

			Implementation of looping constructs and functions
3	14/06/2021	21/06/2021	**PYTHON FOR DATA SCIENCE** • Data Structure • Lists • Dictionaries • Understanding Standard Libraries in Python • Reading a CSV File in Python • Data Frames and basic operations with Data Frames • Indexing Data Frame
4	22/06/2021	29/06/2021	**UNDERSTANDING STATISTICS OF DATA SCIENCE** • Introduction to Statistics • Measures of Central Tendency • Understanding the spread of data • Data Distribution • Introduction to Probability • Probabilities of Discreet and Continuous Variables • Central Limit Theorem and Normal Distribution

5	30/07/2021	05/07/2021	**UNDERSTANDING STATISTICS FOR DATA SCIENCE** • Introduction to Inferential Statistics • Understanding the Confidence Interval and margin of error • Hypothesis Testing • T tests • Chi Squared Tests • Understanding the concept of Correlation
6	06/07/2021	12/07/2021	**PREDICTIVE MODELING** • Introduction to Predictive Modeling • Understanding the types of Predictive Models • Stages of Predictive Models • Data Extraction • Data Exploration • Reading the data into Python • Variable Identification • Univariate Analysis for Continuous Variables • Univar ate Analysis for Categorical Variables • Bivariate Analysis • Treating Missing Values How to treat Outliers • Variable Transformation • Basics of Model Building Linear Regression

			• Logistic Regression • Decision Trees • K-means

CHAPTER 3

TECHNICAL CONTENT

3.1 PYTHON FOR DATA SCIENCE

Python is a general-use high-level programming language that bills itself as powerful, fast, friendly, open, and easy to learn. Python ―plays well with others‖ and ―runs everywhere‖. Python is open source, interpreted, high level language and provides great approach for object-oriented programming. It is one of the best language used by data scientist for various data science projects/application. Python provide great functionality to deal with mathematics, statistics and scientific function. It provides great libraries to deals with data science application. One of the main reasons why Python is widely used in the scientific and research communities is because of its ease of use and simple syntax which makes it easy to adapt for people who do not have an engineering background. It is also more suited for quick prototyping. According to engineers coming from academia and industry, deep learning frameworks available with Python APIs, in addition to the scientific packages have made Python incredibly productive and versatile. There has been a lot of evolution in deep learning Python frameworks and it's rapidly upgrading. In terms of application areas, ML scientists prefer Python as well. When it comes to areas like building fraud detection algorithms and network security, developers leaned towards Java, while for applications like natural language processing (NLP) and sentiment analysis, developers opted for Python, because it provides large collection of libraries that help to solve complex business problem easily, build strong system and data application.

Figure 4 python logo [9]

3.1.1 FEATURES OF PYTHON

1. It uses the elegant syntax, hence the programs are easier to read.
2. It is a simple to access language, which makes it easy to achieve the program working.

3. The large standard library and community support.
4. The interactive mode of Python makes its simple to test codes.
5. In Python, it is also simple to extend the code by appending new modules that are implemented in other compiled language like C++ or C.
6. Python is an expressive language which is possible to embed into applications to offer a programmable interface.
7. Allows developer to run the code anywhere, including Windows, Mac OS X, UNIX, and Linux.
8. It is free software in a couple of categories. It does not cost anything to use or download Pythons or to add it to the application.

3.1.2 MOST COMMONLY USED LIBRARIES FOR DATA SCIENCE:

- **Numpy**: Numpy is Python library that provides mathematical function to handle large dimension array. It provides various method/function for Array, Metrics, and linear algebra. NumPy stands for Numerical Python. It provides lots of useful features for operations on n-arrays and matrices in Python. The library provides vectorization of mathematical operations on the NumPy array type, which enhance performance and speeds up the execution. It‘s very easy to work with large multidimensional arrays and matrices using NumPy.

- **Pandas**: Pandas is one of the most popular Python library for data manipulation and analysis. Pandas provide useful functions to manipulate large amount of structured data. Pandas provide easiest method to perform analysis. It provide large data structures and manipulating numerical tables and time series data. Pandas is a perfect tool for data wrangling. Pandas is designed for quick and easy data manipulation, aggregation, and visualization. There two data structures in Pandas :
  Series – It Handle and store data in one-dimensional data.
  Data Frame – It Handle and store Two dimensional data.

- **Matplotlib**: Matplotlib is another useful Python library for Data Visualization. Descriptive analysis and visualizing data is very important for any organization. Matplotlib provides various method to Visualize data in more effective way. Matplotlib allows to quickly make line graphs, pie charts, histograms, and other professional grade figures. Using Matplotlib, one can customize every aspect of a figure. Matplotlib has interactive features like zooming and planning and saving the Graph in graphics format.

- **Scipy**: Scipy is another popular Python library for data science and scientific Computing. Scipy provides great functionality to scientific mathematics and computing programming. SciPy contains sub-modules for optimization, linear

algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers, Statmodel and other tasks common in science and engineering.

- **Scikit – learn**: Sklearn is Python library for machine learning. Sklearn provides various algorithms and functions that are used in machine learning. Sklearn is built on NumPy, SciPy, and matplotlib. Sklearn provides easy and simple tools for data mining and data analysis. It provides a set of common machine learning algorithms to users through a consistent interface. Scikit

Learn helps to quickly implement popular algorithms on datasets and solve real-world problems.

Figure 5 libraries logo [10]

3.1.3 PYTHON OPERATORS

Operators are used to perform operations on variables and values.

Python divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators

3.1.4 VARIABLE AND DATA TYPE

a. Creating Variables

Variables are containers for storing data values. Unlike other programming languages, Python has no command for declaring a variable. A variable is created the moment you first assign a value to it.

```
x= 5;
y= "John";
print(x); print(y);
```

b. Built-in Data Types

In programming, data type is an important concept. Variables can store data of different types, and different types can do different things.

Python has the following data types built-in by default, in these categories:

- Text Type : Str
- Numeric Types: int, float
- Sequence Types: list, tuple, range
- Mapping Type: Dict
- Set Types: Set
- Boolean Type: Bool

3.1.5 PYTHON CONDITIONS AND IF STATEMENTS

Python supports the usual logical conditions from mathematics:

- Equals: a==b
- Not Equals: a!=b
- Less than: a<b
- Less than or equal to: a<=b
- Greater than: a>b
- Greater than or equal to: a>=b

These conditions can be used in several ways, most commonly in "if statements" and loops.

- An "**if statement**" is written by using the if keyword
  Example :

```
a= 33
b= 200
if b > a;
  print("b is greater than a")
```

- **Elif**

  The elif keyword is pythons way of saying "if the previous conditions were not true, then try this condition".

  Example :

```
a = 33 b
= 33 if
b > a:
print("b is greater than a")
elif a == b:   print("a and b
are equal")
```

- **Else**

  The else keyword catches anything which isn't caught by the preceding conditions.

  Example :

```
a = 200 b
= 33 if b
> a:
print("b is greater than  a")
elif a == b:
print("a and b are equal")
else:
print("a is greater than b")
```

- **Nested If**

  You can have if statements inside if statements, this is called nested if statements.

  Example :

```
x = 41
if x > 10:
print("Above ten,")
if x > 20:
print("and also above 20!")
else:
print("but not above 20.")
```

3.1.6 PYTHON LOOPS

Python has two primitive loop commands:

- while loops
- for loops

A. The while Loop

With the while loop we can execute a set of statements as long as a condition is true.

Example :
Print i as long as i is less than 6:

```
i = 1; while i < 6:    print(i);    i
+= 1;
```

B. For Loops

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).
This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages. With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc.

Example :
Print each fruit in a fruit list:

```
fruits=["apple", "banana", "cherry"] for
x in fruits:
  print(x)
```

3.1.7 PYTHON FUNCTIONS

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result.

- **Creating a Function**

In Python a function is defined using the def keyword:

Example :

```
def  my_function():
  print("Hello from a function")
```

- **Calling a Function**

To call a function, use the function name followed by parenthesis:

Example :

```
def my_function():
print("Hello from a function")
my_function()
```

3.1.8 PYTHON COLLECTION (ARRAYS)

There are four collection data types in the Python programming language:

- **List** is a collection which is ordered and changeable. Allows duplicate members.
- **Tuple** is a collection which is ordered and unchangeable. Allows duplicate members.
- **Set** is a collection which is unordered and unindexed. No duplicate members.
- **Dictionary** is a collection which is unordered, changeable and indexed. No duplicate members

When choosing a collection type, it is useful to understand the properties of that type. Choosing the right type for a particular data set could mean retention of meaning, and, it could mean an increase in efficiency or security.

a) List

A list is a collection which is ordered and changeable. In Python lists are written with square brackets. Create a List: thislist = ["apple", "banana", "cherry"]

```
print(thislist)
```

b) Tuple

A tuple is a collection which is ordered and unchangeable. In Python tuples are written with round brackets.

Create a Tuple: thistuple = ("apple",
"banana", "cherry") print(thistuple)

c) Set

A set is a collection which is unordered and unindexed. In Python, sets are written with curly brackets.

Create a Set: thisset = {"apple",
"banana", "cherry"} print(thisset)

d) Dictionary

A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values.

```
Create and print a dictionary:
thisdict = {   "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict)
```

3.1.9 PANDAS LIBRARY

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Importing pandas library in python :

```
import pandas as pd
```

3.1.10 READ A CSV FILE

Pandas can help read data from different types of files(like .csv and .xlsx)

```
#Reading the data into Python  import
pandas as pd   # import pandas library
df=pd.read_csv(—data.csv‖)  df.head()
df=pd.read_excel (—data.xlsx‖)
df.head()
```

3.1.11 DATA FRAME

A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

Table 1 Structure of data frame

Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

Table 2 Basic operations with Data Frame:

Operations	Command
Dimensions of the dataframe —rows * columns‖	dataframe.shape
Access top <n> records	dataframe.head ()
Access bottom <n> records	dataframe.tail ()
Access all column names	dataframe.columns
Access data of one column	dataframe[—column name‖]
Access data of multiple column	dataframe[—column 1‖, —column 2‖]

3.1.12 INDEXING DATA FRAME

```
#Reading the data into Python  import
pandas as pd   # import pandas library
df=pd.read_csv(—data.csv‖)
# reading the data set
# seeing the dimension of the data set
df.shape
# to see top 5 rows
df.head()
# to see bottom 5 rows
df.tail()
#selecting the name of all column in data frame df.columns
```

```
#selecting the name of single column in data frame
df[—height‖]
#selecting the name of multiple column in data frame
df[—height‖ , —weight‖]
#selecting rows by their positions df.iloc[:5]
#selecting colums by their positions
df.iloc[:,:2]
```

3.2 UNDERSTANDING STATISTIC OF DATA SCIENCE

3.2.1 Introduction to Statistics

Statistics is the discipline that concerns the collection, organization, analysis, interpretation and presentation of data.

3.2.2 Understanding the various variable types

There are two types of variables:

1. Continuous Variables- These are the variables having continuous numerical values
2. Categorical Variables- These are the variables having discrete values.

These are also divided into two parts :

a. Nominal Variables- These are the variables having categories. These do not have an inherent order to them.
b. Ordinal Variables- These are the variables having categories with an order. These have an inherent order to them.

3.2.3 Measures of Central Tendency

A measure of central tendency is a summary statistic that represents the center point or typical value of a dataset. These measures indicate where most values in a distribution fall and are also referred to as the central location of a distribution. In statistics, the three most common measures of central tendency are the mean, median, and mode. Each of these measures calculates the location of the central point using a different method.

a. Mode of the data

The mode is the value that occurs the most frequently in your data set. On a bar chart, the mode is the highest bar. If the data have multiple values that are tied for occurring the most frequently, you have a multimodal distribution. If no value repeats, the data do not have a mode.

Typically, you use the mode with categorical, ordinal, and discrete data. In fact, the mode is the only measure of central tendency that you can use with categorical data.

It is robust and is not generally affected much by addition of a couple of new values.

Syntax :

#Calculating Mode of Gender column mode_data
= data['Gender'].mode()

b. Mean of the data

The mean is the arithmetic average. Mean is not robust. The calculation of the mean incorporates all values in the data. If you change any value, the mean changes.
However, the mean doesn‘t always locate the center of the data accurately.

$$\frac{x_1 + x_2 + \cdots + x_n}{n}$$

Syntax :

#Calculating mean
mean_data = data['Overall Marks'].mean() print(mean_data)

c. Median of the dataset

The median is the middle value. It is the value that splits the dataset in half. To find the median, order your data from smallest to largest, and then find the data point that has an equal amount of values above it and below it.

If dataset is odd then median would be (n+1)/2 th element.
If dataset is even then median would be average of n/2 th and n/2+1 th elements.

Syntax :
#Calculating meadian
median_data = data['Overall Marks'].median() print(median_data)

3.2.4 Outliers in the datasets

Any values will fall really outside the range of the data is termed as as outlier.

Reasons for Outliers in the data are:

1. Typos
2. Measurement error
3. Intentional error
4. Legit outliers

Table 3 Outlier

Without Outlier	With Outlier
4,4,5,5,5,5,6,6,6,7,7	4,4,5,5,5,5,6,6,6,7,7,300
Mean = 5.45	Mean = 30.00
Mode = 5.00	Mode = 5.00

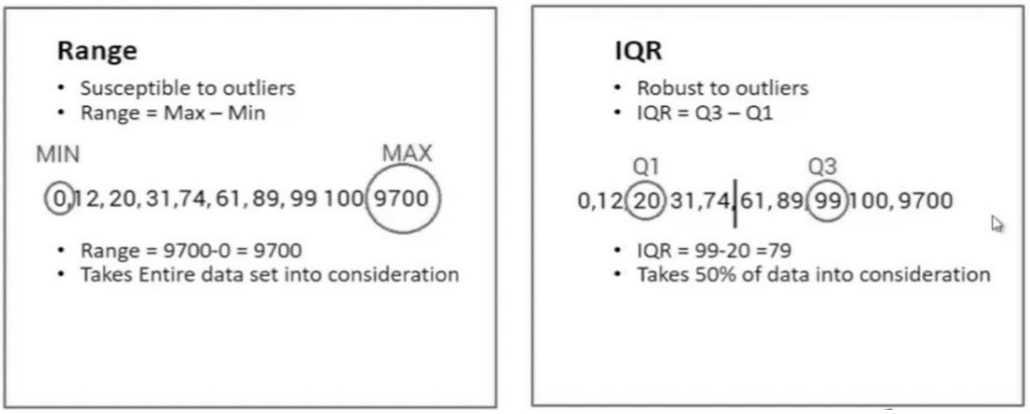
3.2.5 Spread of the data

Spread of the data describes how similar or varied are the set of observations.

The difference between the smallest and largest values is known as Range.

The difference between the third quartile and first quartile values is known as Inter Quartile Range.

Table 4 Range and IQR



Syntax :

```
max_data = data['Overall Marks'].max() min_data
= data['Overall Marks'].min() range_data =
max_data - min_data
print(range_data)
# calculating IQR requires calculating 1st and 3rd quartiles.
Q1 = data['Overall Marks'].quantile(0.25)
Q3 = data['Overall Marks'].quantile(0.75)
IQR = Q3 - Q1 print(IQR)
```

3.2.6 Variance of the data

Variance is the average squared deviation from the mean

$$\sigma^2 = \frac{\sum(\chi - \mu)^2}{N}$$

Syntax :

#Calculating variance
var_data = data['Overall Marks'].var(ddof = 0) print(var_data)

3.2.7 Standard Deviation of the data

The Standard Deviation is a measure of how spreads out numbers are.

Its symbol is **σ**

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

Syntax :

#Calculating standard deviation std =
data['Overall Marks'].std(ddof = 0) std

3.2.8 Frequency Tables

Frequency Table is a table representing frequencies or count of a category.

Syntax :

#Generating frequency table freq_data =
data['Subject'].value_counts()
print(freq_data)

3.2.9 Histograms

Histogram: a graphical display of data using bars of different heights. It is similar to a Bar Chart, but a histogram groups numbers into ranges. The height of each bar shows how many fall into each range.

Syntax :

#Generating histogram import
matplotlib.pyplot as plt
%matplotlib inline
plt.hist(x='Overall Marks',data=histogram) plt.show()

3.2.10 Introduction to Probability

Probability is simply how likely something is to happen. Whenever we're unsure about the outcome of an event, we can talk about the probabilities of certain outcomes—how likely they are. The analysis of events governed by probability is called statistics.

- An experiment or trial is any procedure that can be infinitely repeated and has a well-defined set of possible outcomes, known as the sample space.
- An outcome is a possible result of an experiment or trial.
- An event is a set of outcomes of an experiment (a subset of the sample space) to which a probability is assigned.

3.2.11 Bernoulli Trials and Probability Mass Function

An experiment which has exactly two outcome.
Probability distribution of the number of success in n Bernoulli trials is known as a Binomial distribution.

$$P(X = x) = \frac{n!}{x!(n-x)!}p^x q^{n-x}$$

If p=q=0.5

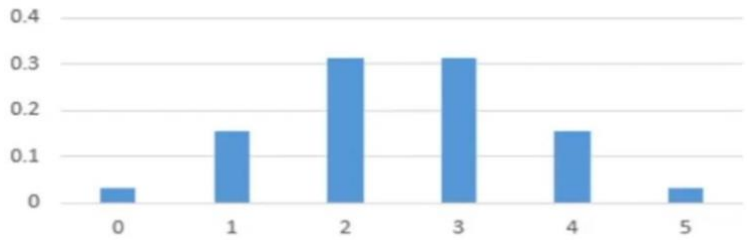


Figure 6 Probability mass function when success and failure is equally likely [11]

For larger number of trails:

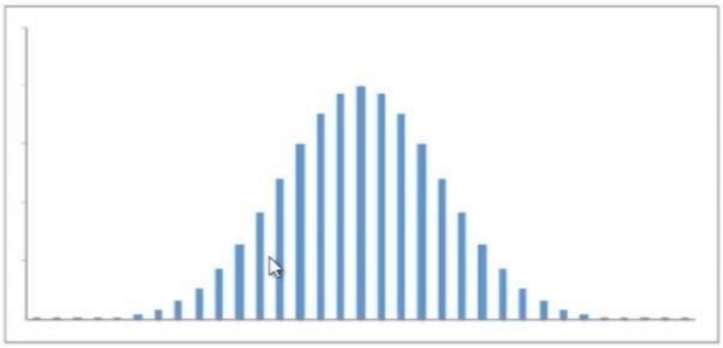


Figure 7 Probability mass function for large number of trails [11]

3.2.12 Probabilities for Continuous Random Variables

Continuous random variables are variables which can take any value in a given range.
For eg:

- Amount of sugar in an orange
- Life of a fly

The Probabilities for Continuous Random Variables are for a range rather than a single value.

3.2.13 The Central Limit Theorem

If we take means of random samples from a distribution and we plot the means, the graph approaches to a normal distribution when we have taken sufficiently large number of such samples.
The theorem also says that the mean of means will be approximately equal to the mean of sample means.

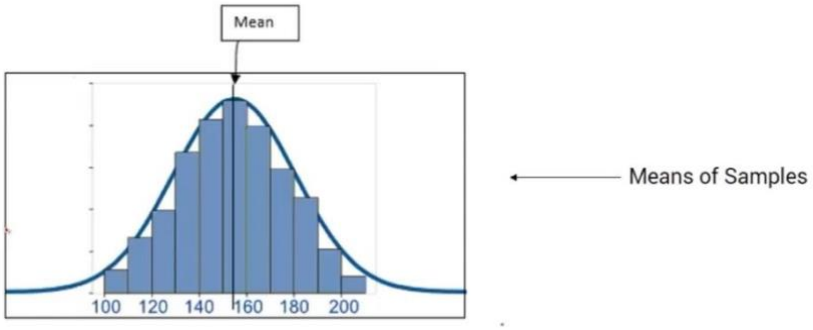


Figure 8 Central limit theorem [11]

3.2.14 Properties of the Normal Distribution

- The distribution is symmetric about the mean
- Normal distribution for higher standard deviation are flatter as compared to those for lower standard deviation.

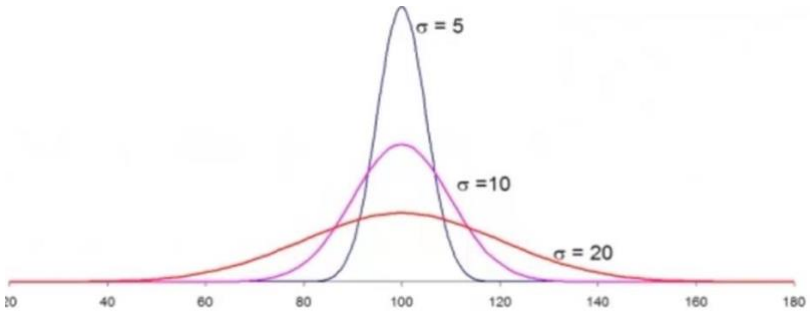


Figure 9 Normal Distribution [12]

Replace frequencies with probabilities.

- Area under the curve would be equal to 1.
- The equation of normal distribution is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

- There is large probabilities for the means to be around the actual mean of the data, than to be father away.

3.2.15 Using the Normal Curve for Calculations

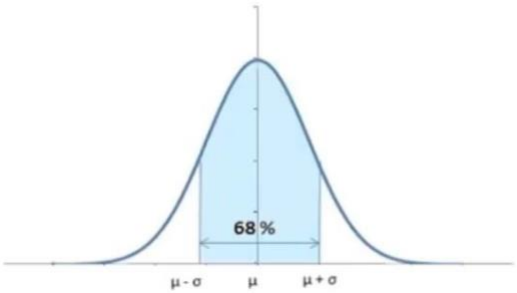


Figure 10 Area around 1st SD [12] Area

around 1st S.D. gives the mean=0.68.

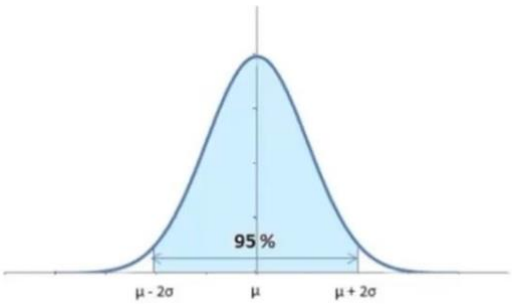


Figure 11 Area around 2nd SD [12] Area

around 2nd S.D. gives the mean= 0.95.

3.2.16 Z scores

- The distance in terms of S.D. the observed value is away from the mean, is the standard score or the Z score.
- A positive Z score indicates that the observed value is Z standard deviation above the mean.
- Negative Z score indicates that the value is below the mean.
- Observed value = $\mu+Z\sigma$

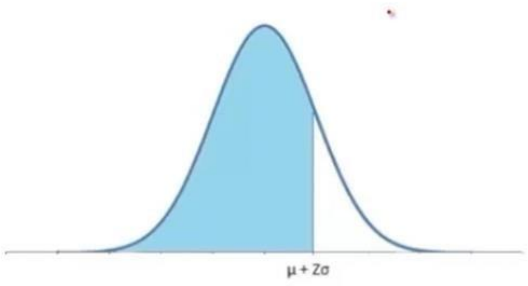


Figure 12 Observed value [13]

3.2.17 Introduction to Inferential Statistics

Making inferences about the population from the sample. Concluding whether a sample is significantly different from the population. Hypothesis testing in general.

Confidence interval defines a region whether there is the highest chances of population statistic to lie. Confidence interval in a given sample is an interval (LB,UP) with k % of confidence level.

$$C.I = \bar{X} \pm Z_{\alpha/2}\, \sigma/\sqrt{n}$$

Margin of error is defines as the sampling error by the person who collected the data. MoE is half of the CI.

3.2.18 Hypothesis Testing

A hypothesis is nothing but a proposed explanation for a phenomenon.

- Null Hypothesis – It can be that sample statistic to be equal to the population statistic or that the intervention doesn‘t bring any difference to the sample.

- Alternate Hypothesis – It basically negates the null hypothesis or says that the intervention brings a significant difference to the sample, or that the sample is significantly different from the population.

Critical value – It is the point (or points) on scale of the test stastic beyond which we reject the null hypothesis and is derived from the level of significance of the test.

Steps to perform Hypothesis Testing

Step 1 : Define Null and Alternate Hypothesis
Step 2 : Set the Decision Criteria
Step 3 : Compute the random chance of probability
Step 4 : Make a Decision

a. Directional and non-directional hypothesis testing

For Directional hypothesis, the null hypothesis is tested in only one direction. In this case one tail test is used.

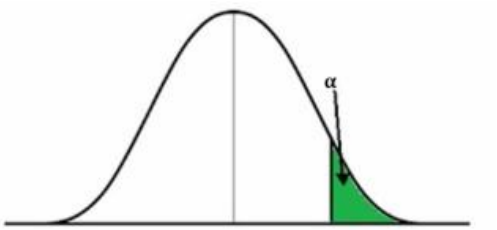


Figure 13 Normal distribution for one tail test [14]

For Non-Directional Hypothesis, the null hypothesis is tested in both the directions. In this case, a two tail test is used.

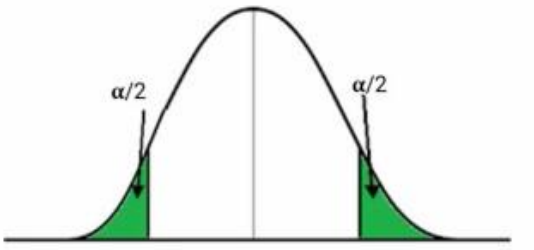


Figure 14 Normal distribution for two tail test [14]

b. Understanding Errors while Hypothesis Testing

	H_0 True	H_0 False
Reject H_0	Type I Error	Correct Rejection
Fail to Reject H_0	Correct Decision	Type II Error

Figure 15 Errors in Hypothesis Testing [15]

Type 1 error – Rejecting the Null hypothesis when it‘s actually true.
Type 2 error – Failing to reject the Null hypothesis when it‘s actually false.

3.2.19 T tests

A t-test is a type of inferential statistic used to determine if there is a significant difference between the means of two groups, which may be related in certain features. A t-test looks at the t-statistic, the t-distribution values, and the degrees of freedom to determine the statistical significance.

The t-test produces two values as its output: t-value and degrees of freedom. The tvalue is a ratio of the difference between the mean of the two sample sets and the variation that exists within the sample sets. Higher values of the t-value, also called tscore, indicate that a large difference exists between the two sample sets. The smaller the t-value, the more similarity exists between the two sample sets.

- A large t-score indicates that the groups are different.
  A small t-score indicates that the groups are similar.

Degree of freedom, df=n-1

a. Conducting One sample T tests

In this there is only one sample.
We find t-test to test if the population with mean μ, that this sample comes from, is significantly different from a population whose mean is μσ.

Steps to do the T tests:

- Define the Null and Alternate Hypothesis
- Compute the T statistic
- Get the T critical values from the tables
- If the t statistic computed is more than t critical value in a positive case, or if the negative t computed is less than the t critical value, we will reject the Null hypothesis.

$$t = \frac{m - \mu}{s/\sqrt{n}}$$

Syntax :

```
#Importing libraries import pandas
as pd import scipy.stats as stats
from scipy.stats import ttest_1samp
#Loading data
data = pd.read_csv("onesample.csv")
data = data['Insect Length'] #Printing
first 5 rows data.head()
t_statistic= ttest_1samp(data, 6.09)
print(t_statistic)
#t-statistic > t-critical We reject the null Hypothesis.
```

b. T-Critical Value

T critical value is calculated using T-table by using degree of freedom , significance level and the type of t test i.e. one tail or two tail.

c. Paired T tests

In paired t test we check if the same sample has changed its behaviour after an intervention, or behaves significantly different in two conditions.

$$t = \frac{\bar{d}}{S_d/\sqrt{n}}$$

d(bar) = mean of the case wise difference between before and after

Syntax:

```
#Importing libraries import
pandas as pd import scipy.stats
as stats from scipy.stats import
ttest_rel
#Loading data
data = pd.read_csv("Data for paired t test.csv") data.head()
# Calculating t and p-value using scipy library
t_statistic, _ = stats.ttest_rel(data['Errors using typewriter'],data['Errors using a
computer'])
#Printing t-statistic t_statistic
# T critical from the table at 0.05 significance level and degree of freedom 24 is 1.711 d.
```

Sample T tests

t = difference/standard error
Difference is the difference in their means and the standard error would be the combine standard.

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}$$

Degree of freedom = N1+N2-2

Syntax:

```
#Importing libraries import
pandas as pd import scipy.stats
as stats from scipy.stats import
ttest_ind
#Loading datasets
data = pd.read_csv('Data for 2 sample test.csv') data.head()
# on referring to the t-table,  t-critical value came out to be 2.056
#Calculating t-statistic and p-value using 2 sample t-test t_statistic=
ttest_ind(data['Defence Colony'],data['Hauz Khas']) t_statistic
# t-statistic> t-critical therefore we reject the null hypothesis.
```

e. Chi Squared Tests

A chi-square statistic is one way to show a relationship between two categorical variables. In statistics, there are two types of variables: numerical (countable) variables and non-numerical (categorical) variables. The chi-squared statistic is a single number that tells you how much difference exists between your observed counts and the counts you would expect if there were no relationship at all in the population.

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

- Oi is the observed frequency
- Ei is the expected frequency
- A low value for chi-square means there is a high correlation between your two sets of data.

You could also use a p-value. First state the null hypothesis and the alternate hypothesis. Then generate a chi-square curve for your results along with a p-value.

Small p-values (under 5%) usually indicate that a difference is significant (or ―small enough‖).

Syntax :

```
#Importing libraries import
pandas as pd import numpy as
np import scipy.stats as stats
from scipy.stats import chisquare
#Importing data
data = pd.read_csv("chi_square.csv")
#Printing first 5 rows data.head()
#Calculating t-statistic and p-value of the chi square test
t_statistic, p_value = chisquare(f_obs= data['Observed'],f_exp=data['Expected']) p_value
```

Hence as p > 0.05, we fail to reject the NULL hypothesis i.e. the observed and expected frequencies are similar.

3.2.20 Correlation

- Correlation is used to determine the relationship between two variables.
- It is denoted by r.
- The value ranges from -1 to 1. Here 0 means no correlation.

$$R = \frac{cov(X,Y)}{Sx,Sy}$$

- The covariance shows how much of these variables vary with each other while the SD shows how much these variables vary apart from each other.

$$Cov(X,Y) = \frac{\Sigma(X-\mu)(Y-\nu)}{n-1}$$

Syntax :

```
data.corr()
#Correlation between two column
data[['Item_MRP','Item_Outlet_Sales']].corr()
```

3.3 PREDICTIVE MODELING

3.3.1 Introduction

Predictive modeling, also called predictive analytics, is a mathematical process that seeks to predict future events or outcomes by analyzing patterns that are likely to forecast future results. Once data has been collected, the analyst selects and trains statistical models, using historical data. Although it may be tempting to think that big data makes predictive models more accurate, statistical theorems show that, after a certain point, feeding more data into a predictive analytics model does not improve accuracy. In many use cases, including weather predictions, multiple models are run simultaneously and results are aggregated to create one final prediction. This approach is known as ensemble modeling. As additional data becomes available, the statistical analysis will either be validated or revised.

3.3.2 Applications of Predictive Modeling

One of the most common uses of predictive modeling is in online advertising and marketing. Modelers use web surfers' historical data, running it through algorithms to determine what kinds of products users might be interested in and what they are likely to click on.Bayesian spam filters use predictive modeling to identify the probability that a

given message is spam. In customer relationship management predictive modeling is used to target messaging to customers who are most likely to make a purchase.

3.3.3 Steps involved in Predictive analysis

1 Data exploration

2 Data cleaning
3 Modelling

3.3.4 Data exploration

Data exploration is the initial step in data analysis, where users explore a large data set in an unstructured way to uncover initial patterns, characteristics, and points of interest. This process isn‘t meant to reveal every bit of information a dataset holds, but rather to help create a broad picture of important trends and major points to study in greater detail.Data exploration can use a combination of manual methods and automated tools such as data visualizations, charts, and initial reports.

This process makes deeper analysis easier because it can help target future searches and begin the process of excluding irrelevant data points and search paths that may turn up no results. More importantly, it helps build a familiarity with the existing information that makes finding better answers much simpler. Many times, data exploration uses visualization because it creates a more straightforward view of data sets than simply examining thousands of individual numbers or names. In any data exploration, the manual and automated aspects also look at different sides of the same coin. Manual analysis helps users familiarize themselves with information and can point to broad trends. These methods are also by definition unstructured so that users can examine a whole set without any preconceptions. Automated tools, on the other hand, are excellent at pruning out less applicable data points, reorganizing data into sets that are easier to analyse, and scrubbing data sets to make their findings relevant.

Steps performed during data exploration – **Syntax:**

```
#Reading the data into Python
#To read the data in python following line of code used import
pandas as pd   # import pandas library
df=pd.read_csv(—data.csv)
# reading the data set
# seeing the dimension of the data set df.shape
# to see top 5 rows df.head()
# to see bottom 5 rows  df.tail()
```

3.3.5 Data cleaning

Data Cleaning means the process of identifying the incorrect, incomplete, inaccurate, irrelevant or missing part of the data and then modifying, replacing or deleting them according to the necessity. Data cleaning is considered a foundational element of the basic data science. Data is the most valuable thing for Data science. In computing or Business data is needed everywhere. When it comes to the real world data, it is not improbable that data may contain incomplete, inconsistent or missing values. If the data is corrupted then it may hinder the process or provide inaccurate results. Let‘s see some examples of the importance of data cleaning. Suppose you are a general manager of a company. Your company collects data of different customers who buy products produced by your company. Now you want to know on which products people are interested most and according to that you want to increase the production of that product. But if the data is corrupted or contains missing values then you will be misguided to make the correct decision and you will be in trouble.

Data cleaning involves various methods:

(a) Variable identification
(b) Treating missing value
(c) Treating outliers

1. **Variable identification**

Variable identification or analysis can be done in two ways, univariate analysis, and bivariate analysis.

- **Univariate analysis:** Univariate analysis is used to highlight missing and outlier values. Here each variable is analysed on its own for range and distribution. Univariate analysis differs for categorical and continuous variables. For categorical variables, you can use frequency table to understand distribution of each category. For continuous variables, you have to understand the central tendency and spread of the variable. It can be measured using mean, median, mode, etc. It can be visualized using box plot or histogram.

  Syntax:
  Univariate analysis for continuous variables

```
import pandas as pd   # import pandas library
df=pd.read_csv(_data.csv‘)  # reading the data set
# seeing the dimension of the data set
df.shape
# seeing the variable of columns
df.columns # to
see top 5 rows
df.head()
```

Variable identification
df.types
Univariate analysis df.describe() % use to calculate mean count maximum value and standard deviation.
#plotting a histogram for age variable
df[_Age‘].plot.hist()

Univariate analysis for categorical variables #
creating frequency table for categorical variable Sex
df[_Sex‘].value_count()
#create percentages from frequency
df[_Sex‘].value_count()/len(df[_Sex‘])
creating a bar plot for sex variable
df[_Sex‘].value_count().plot.bar()

- **Bivariate analysis :** Bivariate analysis is the simultaneous analysis of two variables (attributes). It explores the concept of relationship between two variables, whether there exists an association and the strength of this association, or whether there are differences between two variables and the significance of these differences.

There are three types of bivariate analysis.

1. Continuous – Continuous analysis
2. Categorical – Continuous analysis
3. Categorical – Categorical analysis

a) Continuous – Continuous analysis

While doing bi-variate analysis between two continuous variables, we should look at scatter plot. It is a nifty way to find out the relationship between two variables. The pattern of scatter plot indicates the relationship between variables. The relationship can be linear or non-linear. Scatter plot shows the relationship between two variable but does not indicates the strength of relationship amongst them. To find the strength of the relationship, we use Correlation. Correlation varies between -1 and +1.

(a)-1: perfect negative linear correlation
(b)+1: perfect positive linear correlation and (c)
0: No correlation

Syntax : import pandas as pd # import pandas
library df=pd.read_csv(_data.csv‘) # reading
the data set
seeing the dimension of the data set df.shape

```
# seeing the variable of columns
df.columns # to
see top 5 rows
df.head()
# Variable identification df.types
# Plotting scatter plot
df.plot.scatter(_Age‘,_Fare‘)
# creating the correlation
df.corr()
```

b) Categorical – Continuous analysis

While exploring relation between categorical and continuous variables, we can draw box plots for each level of categorical variables. If levels are small in number, it will not show the statistical significance. To look at the statistical significance we can perform Z-test, T-test.

Syntax:

```
file.groupby(_Sex‘)[_Age‘].mean()
file.groupby(_Sex‘)[_Age‘].mean().plot.bar
# Importing the scipy library for t-test
males=file[file[_Sex‘]== _male‘]
females=file[file[_Sex‘]== _female‘]
ttest_ind(males[_Age‘],females[_Age‘])
```

c) Categorical – Categorical analysis

It is used to visualize the relationship between two categorical variables. It compares the percentage that each category from one variable contributes to a total across categories of the second variable.

Syntax:

```
pd.crosstab(file[_Sex‘], file[_Supervised‘]) from
scipy.stats import chi2_contiggency
chi2contingency(pd.crosstab(file[_Sex‘],file[_Survived‘])
)
```

2. **Treating missing value**

Reasons for missing values:

- Non –response –for example when you collect data people‘s income and many choose not to answer

- Error in Data collection
- Error in reading Data

Different methods to deal with missing values

1. Imputation: This works by calculating the mean/median of the non-missing values in a column and then replacing the missing values within each column separately and independently from the others. It can only be used with numeric data.
2. Deletion: There are two ways either row wise deletion or either column wise deletion.

Syntax: import pandas as
pd
file=pd.read_csv(—data.csv
")
file.shape file.describe()
file.isnull()
file.isnull().sum()
# dropping all rows wherever there are any missing values file.dropna()
#dropping rows where all the entries are missing
file.dropna(how=_all‘)
#dropping column with any missing value
file.dropna(axis=1)
# filling all the missing values in a data frame with 0 file.filena(0)

3. **Treating outliers:-**Outliers are extreme values that deviate from other observations on data, they may indicate a variability in a measurement, experimental errors or a novelty. In other words, an outlier is an observation that diverges from an overall pattern on a sample.

Most common causes of outliers on a data set:

- Data entry errors (human errors)
- Measurement errors (instrument errors)
- Experimental errors (data extraction or experiment planning/executing errors)
- Intentional (dummy outliers made to test detection methods)
- Data processing errors (data manipulation or data set unintended mutations)
- Sampling errors (extracting or mixing data from wrong or various sources)
- Natural (not an error, novelties in data

Types of outliers

- Univariate

- Bivariate

Treating Outliers

1. Deleting Observations
2. Transforming and Binning values
3. Imputing Outliers like missing values
4. Treat them as separately

Syntax:

```
Import pandas as pd
Import matplotlib.pyplot as ply
%matplotlib inline
df=pd.read_csv(―data.csv‖)
df.head()
# Univariate Outlier detection
#creating age box plot
df[_Age‘].plot.box() # Bivariate
Outlier detection # creating scatter
plot for age and fare
df.plot.scatter(_Age‘, _Fare‘) #
Removing outliers
df-=df[df[_Fare‘]<300]
```

Variable Transformation

In variable transformation we replace a variable with some function of that variable. For example replacing a variable x with its logarithm. We change the distribution or relationship of a variable with others.

Common methods-

- Logarithm
- Square root
- Cube root
- Binning

Syntax :

```
Import pandas as pd
Import matplotlib.pyplot as ply
%matplotlib inline Import
numpy as np
df=pd.read_csv(―data.csv
```

‖) df.head()
df[_Age‘].plot.hist()
# applying log function
np.log(df[_Age‘]).plot.hist()
np.sqrt(df[_Age‘]).plot.hist()

3.3.6 Modelling

It is a process to create a mathematical model for estimating the future behaviour based on past data. For example a retail bank wants to know the default behaviour of its credit card customers. They want to predict the probability of default for each customer with in next 3 month.

Algorithm used in Model Building are following given below

1. Linear Regression

A linear regression model is a linear approximation of a causal relationship between two or more variables. ... Y is the variable we are trying to predict and is called the dependent variable. X is an independent variable. When using regression analysis, we want to predict the value of Y, provided we have the value of X.

Syntax for Linear Regression:

import pandas as pd import
numpy as np import
matplotlib.pyplot as plt
%matplotlib inline # Reading the dataset
data=pd.read_csv(_train.csv‘) data.shape
data.head() #making test dataset
test=data[8000:] test.head()
x_train=train.drop(_Item_Outlet_Sales‘,axis=1
) y_train=train[_Item_Outlet Sales‘,axis=1]
true_p=test[_Item_Outlet_Sales‘]
from sklearn.linear_model import LinearRegression
1reg=LinearRegression()
1reg.fit(x_train,y_train)
x_train=pd.get_dummies(x_train)
x_train.fillna(0,inplace=True)
x_train.fillna(0,inplace=True)
pred=1reg.predict(x_test) #performance
of our model
1regscore(x_test,true_p)
1reg.score(x_train,y_train)
rmse_test=np.sqrt(np.mean(np.power((np.array(true_p)-np.array(pred)),2)))

rmse_train=np.sqrt(np.mean(np.power((np.array(y_train)np.array(1reg.predict(x_train))),2))) print(rmse_test)
print(rmse_train)

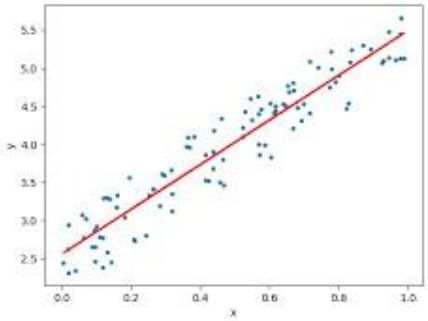


Figure 16 Linear Regression [16]

2. Logistic Regression

Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a data set. ... Based on historical data about earlier outcomes involving the same input criteria, it then scores new cases on their probability of falling into a particular outcome category.

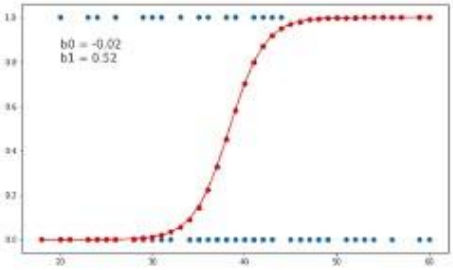


Figure 17 Logistic Regression [17] **Syntax**

for Logistic Regression :

import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
df = pd.read_csv("insurance_data.csv") df.head()
plt.scatter(df.age,df.bought_insurance,marker='+',color='red') from
sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =
train_test_split(df[['age']],df.bought_insurance,train_size=0.8) X_test
from sklearn.linear_model import LogisticRegression
model = LogisticRegression() model.fit(X_train,
y_train) X_test y_predicted = model.predict(X_test)

model.predict_proba(X_test)
model.score(X_test,y_test)
y_predicted
X_test
#model.coef_ indicates value of m in y=m*x + b equation model.coef_
model.intercept_ indicates value of b in y=m*x + b equation model.intercept_
#Lets defined sigmoid function now and do the math with hand
import math def sigmoid(x): return 1 / (1 + math.exp(-x)) def
prediction_function(age): z = 0.042 * age - 1.53 # 0.04150133 ~
0.042 and -1.52726963 ~ -1.53 y = sigmoid(z) return y age = 35
prediction_function(age) age
= 43
prediction_function(age)

3. Decision Trees

A Decision Tree is an algorithm used for supervised learning problems such as classification or regression. A decision tree or a classification tree is a tree in which each internal node is labelled with an input feature. The arcs coming from a node labelled with a feature are labelled with each of the possible values of the feature. Each leaf of the tree is labelled with a class or a probability distribution over the classes. A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions. This process of top-down induction of decision trees is an example of a greedy algorithm, and it is the most common strategy for learning decision trees.

Syntax for Decision Trees:

import pandas as pd import
numpy as np
data=pd.read_csv('data_cleaned.csv')
data.shape data.head()

#seperating independent and dependent variables
x = data.drop(['Survived'], axis=1) y =

data['Survived'] from sklearn.model_selection

import train_test_split

train_x,test_x,train_y,test_y = train_test_split(x,y, random_state = 101, stratify=y)
train_y.value_counts()/len(train_y) test_y.value_counts()/len(test_y)

#importing decision tree classifier from
sklearn.tree import DecisionTreeClassifier clf =
DecisionTreeClassifier() clf.fit(train_x,train_y)
clf.score(train_x, train_y) clf.score(test_x,
test_y) clf.predict(test_x)

4. K-means

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. ... In other words, the K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible

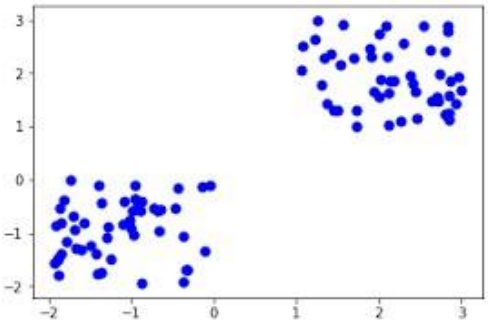


Figure 18 K Mean [18]

Syntax for K-means: import pandas as pd

import numpy as np
from matplotlib.pyplot import plot as plt
%matplotlib inline from sklearn.cluster
import KMeans
data=pd.read_csv("student_evaluation.csv")
data.shape data.head() pd.isnull(data).sum()
data.describe() kmeans =
KMeans(n_clusters=2) kmeans.fit(data)
pred=kmeans.predict(data) pred
pd.Series(pred).value_counts() kmeans.inertia_
kmeans.score(data) SSE = [] for cluster in range(1,20):
kmeans = KMeans(n_jobs = -1, n_clusters = cluster)
kmeans.fit(data)
SSE.append(kmeans.inertia_)
frame = pd.DataFrame({'Cluster':range(1,20), 'SSE':SSE})
plt.figure(figsize=(12,6)) plt.plot(frame['Cluster'],
frame['SSE'], marker='o') from
sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

```
data_scaled = scaler.fit_transform(data)
pd.Dataframe(data_scaled).describe() SSE_scaled = []
for cluster in range(1,20):     kmeans = KMeans(n_jobs
= -1, n_clusters = cluster)
kmean.fit(data_scaled)
SSE.append(kmeans.inertia_)
frame_scaled = pd.Dataframe({'Cluster':range(1,20), 'SSE':SSE_scaled})
plt.plot(frame_scaled['Cluster'], frame_scaled['SSE'], marker='o')
plt.xlabel("Clusters") plt.ylabel("SSE")
kmeans = KMeans(n_jobs = -1, n_clusters = 4)
kmean.fit(data_scaled) pred =
kmeans.predict(data_scaled) pred
frame = pd.DataFrame(data_scaled) frame['cluster']
= pred frame.loc[frame['cluster']==2,:]
```

3.4 PROJECT: Covid-19 Data Analysis

Objective: - To perform the data analysis of COVID-19 in India.

Software Requirements: - Jupyter notebook, datasheet file of COVID-19

Description: - In this project we are going to analyse the data of COVID-19. As we all know that coronavirus is a harmful disease so in this project we are going to explain or find the relationship between various parameters such as the relationship between confirmed and cured cases, etc.

Dataset Source: - https://www.kaggle.com/sudalairajkumar/covid19-in-india

Code: -

```
import pandas as pd   import numpy as np   import matplotlib.pyplot
as plt


#reading the data set data=pd.read_csv(—C:\Users\DELL\Downloads\
COVID19_line_list_data.csv`)


# seeing the dimension of the data set data.shape


# to see top 5 rows data.head()
# to see bottom 5 rows
data.tail()


# to see columns data.columns


# to calculate mean ,count etc data.describe()
```

# to see null values
data.isnull().sum()

#relating the varibles with scatterplots
# analyzing the relation between confirmed and cured cases
sns.relplot(x=—confirmed,y=—cured,data=data)

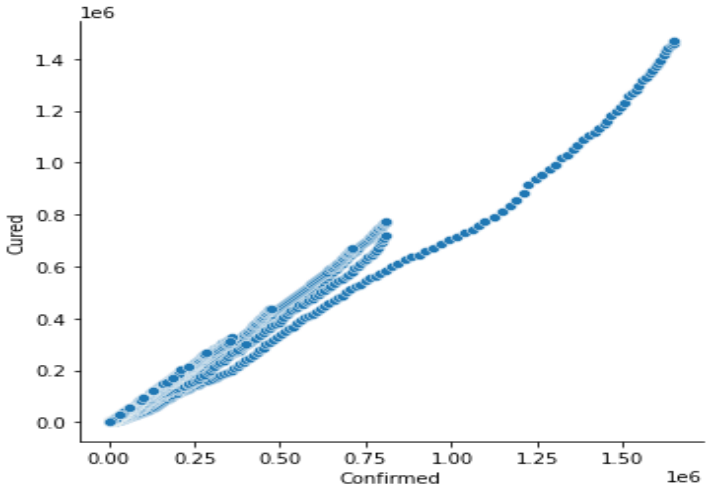


Figure 19 Output of relation between confirmed and cured cases [4]

# analysing the relation between confirmed and death cases
sns.relplot(x=—confirmed,y=—death,data=data)

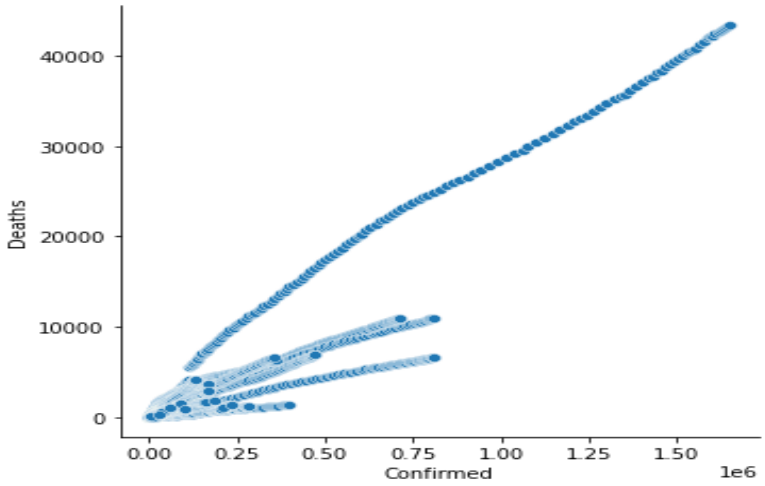


Figure 20 Output of relation between confirmed and cured cases [4]

analysing the relation between all the columns in pair sns.pairplot(data)

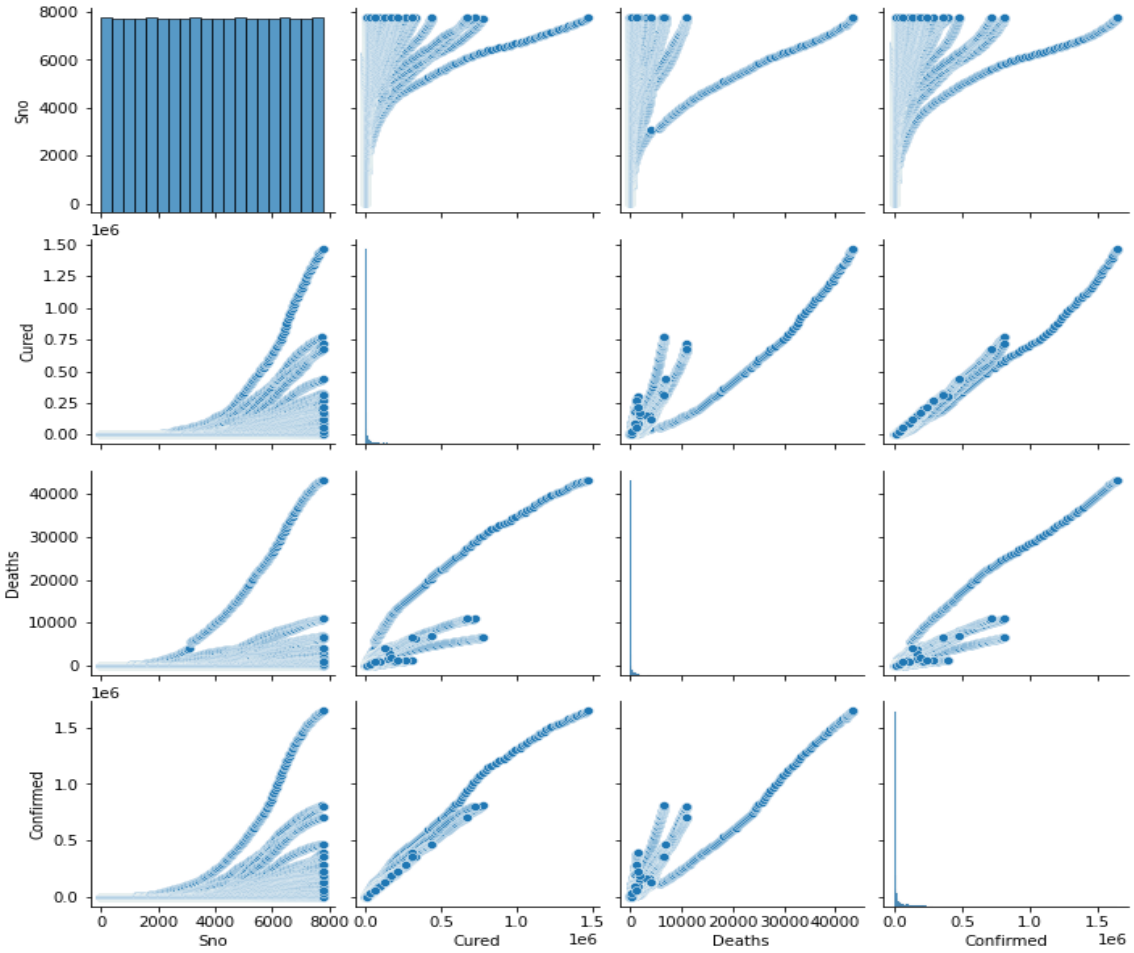


Figure 21 Output of the relation between all the columns in pair [4]

Result- The data analysis of COVID-19 is successfully implemented in jupyter notebook.

CHAPTER 4

FINDINGS

After the completion of 6 weeks training in Data Science, we are able to:

- RECALL – Recall the topics like :- Basic of python and basic of statistics (which includes mean, mode , median, variance, etc.).
- UNDERSTAND – Understand the topic of Data Science which includes various steps such as data extraction, data cleaning and applying appropriate model by which we can make future predictions.
- APPLY – After understanding the concept of Data Science we are able to apply the data science concept to various problems.

CONCLUSION

Data science is a "concept to unify statistics, data analysis and their related methods" in order to "understand and analyse actual phenomena" with data. It uses techniques and theories drawn from many fields within the context of mathematics, statistics, computer science, domain knowledge and information science.

Big data is very quickly becoming a vital tool for businesses and companies of all sizes. The availability and interpretation of big data has altered the business models of old industries and enabled the creation of new ones.

Data scientists are responsible for breaking down big data into usable information and creating software and algorithms that help companies and organizations determine optimal operations. The field encompasses analysis, preparing data for analysis, and presenting findings to inform high-level decisions in an organization.

In our training we get to know about the basic & advanced techniques for understanding and analysing data.

In our training we used python for data science as it is commonly used for data science. It is a programming language with simple syntax and there are a number of python libraries that are used in data science including numpy, pandas, and scipy.

Now, we are able to acquire the confidence to deal with problems that arise in data science projects.

During the period of training the Internshala helped us a lot and we are provided with the online material to study data science in this time of pandemic. So I am grateful to Internshala and my teachers which made this training possible.

I hope this experience will surely help me in my future and also in shaping my career.

REFERENCES

[1]Kim T. K. ―T test as a parametric statistic, Korean Journal of Anesthesiology,2017; 2005-7563

[2]Lee DK, In J., Lee S., ―Standard deviation and standard error of the mean, Journal of Anesthesiology 2015; 68: 220-3

[3]Wikipedia, ―Data science, https://en.wikipedia.org/wiki/Data_science

[4]Dataset on Novel Corona Virus Disease 2019 in India, ―KAGGLE, https://www.kaggle.com/sudalairajkumar/covid19-in-india

[5]David Barber, ―Bayesian Reasoning and Machine Learning, 2012

[6]https://intellipaat.com/blog/what-is-data-science

[7]https://www.javatpoint.com/data-science

[8]https://dataimpact.co.uk/2016/03/13/data-science-process/

[9]https://github.com/python/cpython

[10]https://www.analyticsvidhya.com/blog/2019/07/dont-miss-out-24-amazing-pythonlibraries-data-science/

[11]https://www.analyticsvidhya.com/blog/tag/probability-density-function/

[12]https://www.analyticsvidhya.com/blog/2020/04/statistics-data-science-normaldistribution/

[13]https://www.analyticsvidhya.com/blog/tag/z-score/

[14]https://statisticsbyjim.com/hypothesis-testing/t-tests-t-values-t-distributionsprobabilities/

[15]https://www.sixsigma_institute.org/Six_Sigma_DMAIC_Process_Analyze_Phase_Hypothesis_Testing.php

[16]https://en.m.wikipedia.org/wiki/Linear_regression

[17]https://en.m.wikipedia.org/wiki/Logistic_regression

[18]https://en.m.wikipedia.org/wiki/K-means_clustering